

10-03-00

A

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)*(Only for new nonprovisional applications under 37 CFR 1.53(b))*Docket No.
POU920000162US1

Total Pages in this Submission

TO THE ASSISTANT COMMISSIONER FOR PATENTSBox Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

WRITE-BEFORE-READ INTERLOCK FOR R-UNIT OPERANDS

and invented by:

Scott B. Swaney et al.

If a CONTINUATION APPLICATION, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 12 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☐ Cross References to Related Applications (if applicable)
 - c. ☐ Statement Regarding Federally-sponsored Research/Development (if applicable)
 - d. ☐ Reference to Microfiche Appendix (if applicable)
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings (if drawings filed)
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
POU920000162US1

Total Pages in this Submission

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*

a. ☒ Formal Number of Sheets 1

b. ☐ Informal Number of Sheets _____

4. ☒ Oath or Declaration

a. ☒ Newly executed *(original or copy)* ☐ Unexecuted

b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*

c. ☒ With Power of Attorney ☐ Without Power of Attorney

d. ☐ DELETION OF INVENTOR(S)

Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*

The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied
under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby
incorporated by reference therein.

6. ☐ Computer Program in Microfiche *(Appendix)*

7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*

a. ☐ Paper Copy

b. ☐ Computer Readable Copy *(identical to computer copy)*

c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☒ Assignment Papers *(cover sheet & document(s))*

9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*

10. ☐ English Translation Document *(if applicable)*

11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

☐ First Class ☒ Express Mail *(Specify Label No.):* **EK954544668US**

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
POU920000162US1

Total Pages in this Submission

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☐ Additional Enclosures *(please identify below):*

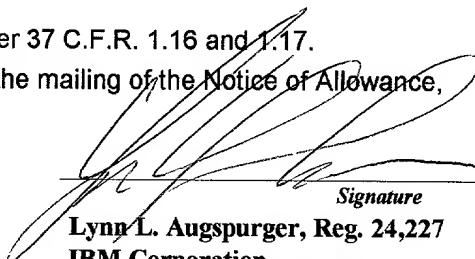
Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	22	- 20 =	2	x \$18.00	\$36.00
Indep. Claims	2	- 3 =	0	x \$78.00	\$0.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$690.00
OTHER FEE <i>(specify purpose)</i>					\$0.00
TOTAL FILING FEE					\$726.00

- ☐ A check in the amount of _____ to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **09-0463** as described below. A duplicate copy of this sheet is enclosed.
- ☒ Charge the amount of **\$726.00** as filing fee.
 - ☒ Credit any overpayment.
 - ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
 - ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: **October 2, 2000**


Signature
Lynn L. Augspurger, Reg. 24,227
IBM Corporation

Intellectual Property Law
2455 South Road, P386
Poughkeepsie, New York 12601
(845) 433-1174
Fax: (845) 432-9786

CC:

Docket No: POU920000162US1

WRITE-BEFORE-READ INTERLOCK
FOR R-UNIT OPERANDS

APPLICATION FOR
UNITED STATES LETTERS PATENT

Express Mail Label No: EK954544668US

Date of Deposit: October 2, 2000

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee Service under 37 CFR 1.10 on the date indicated above and is addressed to Box Patent Application, Commissioner of Patents and Trademarks, Washington, D. C. 20231.

Susan L. Nelson

Susan L. Nelson

I N T E R N A T I O N A L B U S I N E S S M A C H I N E S
C O R P O R A T I O N

0957463-100300

WRITE-BEFORE-READ INTERLOCK FOR R-UNIT OPERANDS

BACKGROUND OF THE INVENTION

The present invention relates generally to detecting the updating of recovery unit (R-unit) registers and in particular to a write-before-read interlock that returns data as soon as it is available. A detailed description of a recovery unit for mirrored processors is found in U.S. Patent No. 5,692,121 to Bozso et al., issued November 25, 1997, and incorporated by reference.

Currently, high performance processors are typically pipelined. Pipelining is a method of processing that allows for fast concurrent processing of data. This is accomplished by overlapping operations in a portion of memory that passes information from one process to another process. Most processor instructions have to go through the same basic sequence: the instruction must be fetched, it must be decoded, it must have its operands fetched, (which may first require address generation), it must be executed, and its results must be put away. The pipelining method fetches and decodes instructions in which, at any given time, several program instructions are in various stages of being fetched or decoded. The pipelining method improves the speed of system execution time by ensuring that the microprocessor does not have to wait for instructions. When the processor completes execution of one instruction, the next instruction is ready to be performed.

Computer systems contain several types of registers including General Purpose registers (GPRs), Access registers (ARs), Floating Point registers (FPRs), Host and Guest Control registers (CRs). The millicode architecture contains additional registers - Millicode General Purpose registers (MGRs), Millicode Access registers (MARs),

Millicode Control registers (MCRs), and System registers (SYSRs). These millicode registers may only be directly accessed by millicode instructions. There are multiple copies of some of these registers throughout the processor, for performance and cycle-time reasons. When an instruction is executed which updates one of these registers, the
5 new value is placed on a common bus "C-bus" during the put-away stage of the pipeline, and all copies of the registers are updated from the common bus. In order to be able to recover from errors, one copy of every register is contained in the R-unit, and protected by an Error Correction Code (ECC). All the registers in the R-unit are assigned an 8-bit address. In order to recover from errors, all of these registers, and storage, must be
10 "check-pointed". This means that at any given time, there must be one copy of the registers and storage that reflect the results at the completion of an instruction. In other words, if an instruction updates multiple registers, or a register and storage, either all of the updates must be visible, or none of them. When an error is encountered, all copies of the registers are restored to their check-pointed state, and return control back to the point
15 following the last instruction to complete. The R-unit copy of the registers is the check-pointed copy. Ensuring that the register updates are synchronous with any storage updates requires additional machine cycles from when the results are "put-away" until they are check-pointed. GPRs, ARs, FPRs, MGRs, and MARs are used frequently, so they must be accessible without incurring any stalls in the pipeline, or performance would
20 suffer. Thus, the execution unit contains copies of these registers so that they may be accessed locally without incurring a pipeline stall. CRs and MCRs are not used frequently, so performance is not significantly impacted if a pipeline stall is encountered while accessing them. Thus, the only copy of these registers is the ECC protected copy in the R-unit. Because of the additional cycles after the put-away cycle required to maintain

the R-unit checkpoint, access to registers from the R-unit may incur pipeline stalls.

Typically, these R-unit operands are not used or updated until late in the pipeline. This results in the loss of processor time due to unnecessary clock cycles. In addition, instructions in the pipeline may update an R-unit register that is a source operand used in a later instruction (pre-fetching) for its operands. Again, this condition is not efficient and results in added clock cycles. Consequently, there is a need for a quicker and more efficient method of detecting when R-unit registers are being updated during the pre-fetching of operands period to guarantee that the correct values are sent for access to register operands in the R-unit with a minimum number of clock cycles required.

10 BRIEF SUMMARY OF THE INVENTION

An exemplary embodiment of the invention is a method for holding up R-unit operands by comparing R-unit register addresses in at least one queue and interlocking valid address matches. The method includes receiving a plurality of R-unit register addresses and storing these R-unit register addresses in at least one queue. This method includes a write queue, a read queue, and a pre-write queue. Further, this method requires accessing these queues and comparing the R-unit register addresses therein. After the addresses are compared the method determines whether there is a valid match between the R-unit register addresses and if so, implementing one or more interlocks.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings, wherein like elements are numbered alike in the FIGURE:

FIG.1 is a system diagram for implementing the write-before-read interlock

feature in one preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

As discussed herein, current computer systems have micro-processor pipelines which include R-unit register operands that are accessed early in the pipeline, but are not
5 used or updated until late in the pipeline. An instruction in the pipeline may update an R-unit register that is a source operand for a later instruction that is pre-"fetching" its operands.

The present invention solves the problem of accessing R-unit register operands which have updates pending from instructions further along in the pipeline. Specifically,
10 this problem is solved by introducing a write-before-read interlock that detects this condition and waits until all updates to the register have been made before returning its value, incurring a minimum number of penalty cycles.

In addition, the R-unit uses three queue structures that compare addresses to implement the write-before-read interlock (i.e. read queue, write queue, pre-write queue).
15 With the interlock, data is returned for the operand as soon as it is available. Without the interlock, an interface would have to be added between the R-unit and decode device to hold up (or reject) decode operations until there are no more pending updates to the source register, which could result in additional penalty cycles and negatively impact performance. Therefore, the present invention saves valuable processing time.

FIG. 1 is an exemplary system diagram for the preferred embodiment of the
20 invention for returning R-unit register operand data as soon as it becomes available. Copies of all architected and semi-architected registers are maintained in the R-unit, protected by ECC for recovery purposes. The entire register space is mapped to an 8-bit

R-unit address. Most of the registers in the R-unit (addresses '00'x-'9F'x) may only be updated via a common data bus (C-bus) 24 from an execution unit 36. Since these registers only have one update source, they are implemented in a single port SRAM 32. The remaining registers in the R-unit are updated by separate hardware interfaces as well
5 as the C-bus 24, and are implemented as discrete registers.

For instructions that update multiple registers, the partial results must be accumulated until the instruction completes, then all partial results are check-pointed on the same cycle. For the addresses that are implemented as discrete registers, there is a fixed timing from the time the execution unit 36 "endops" the instruction until the register
10 values are check-pointed in the R-unit and available to be read. However, for the addresses that are implemented in the single-port SRAM 32, partial results are accumulated and check-pointed in the write queue 26 with the same fixed timing. However, the accumulated partial results are not available to be read from the R-unit until they drain from the write queue into the single-port SRAM 32 over an undetermined
15 number of cycles. Consequently, the number of cycles depends on how many partial results were required for the instruction, as well as, whether or not subsequent instructions required access to the SRAM 32 for source operands. (Priority is given to read the single-port SRAM 32, unless the address being read has a pending update.)

For instructions which require R-unit source operands, or explicitly update R-unit
20 registers, decode device 12 sends a request to the R-unit containing the R-unit address, and predetermined bits to indicate whether the corresponding address is to be read, written, or both. These requests are put into the read queue 14 (or bypass device 16 if the read queue 14 is empty). Since the micro-processor decodes and executes instructions in-order, the read queue 14 is implemented FIFO, so that the requests are processed

serially in the order in which they were received from the decode device 12.

When a valid read address is presented, either from the read queue 14 or bypassed from the decode device 12, the address is compared, with the comparator 30, against all valid entries in the write queue 26. If the read address matches any valid address in the write queue 26, then the read cannot be processed and a "block read" condition is set to indicate such. During the block read the control 28 switches the priority for the single SRAM 32 port from processing read addresses from the read queue to draining the accumulated results from the write queue 26 until the address compare is no longer active. This piece of the interlock guarantees that any register updates already sent by the execution unit 36 will be properly reflected in the data returned for the read address. Note that while reads are being blocked, the decode device 12 may still send more requests to the read queue 14. If the read queue 14 approaches full, it must block the decode device 12. This has a negligible performance impact, as it means the decode device 12 is far ahead of execution.

The second part of the interlock covers the cases where an instruction which updates an R-unit register has not reached the put-away stage of the pipeline yet. These instructions update an R-unit register that contains a source operand used for a later instruction that has already decoded. This interlock compares the read address (r)18 against the entries in the pre-write queue 22, that contain register addresses which are going to be written.

For instructions that explicitly update R-unit registers, the decode device 12 turns on a "write" (w) bit in the request to the R-unit. When this request is processed from the read queue 14, the write bit tells the control 20 to put this request into the pre-write queue 22. If the read bit is also on, then the request will not move to the pre-write queue 22.

until the actual read is processed (no block read condition). If only the write bit is on, then the request moves directly to the pre-write queue 22. When a register is updated via the C-bus 24, the address is compared against the address of the oldest entry in the pre-write queue 22. If the address matches, then the entry is removed from the pre-write
5 queue 22. Although unlikely, it is theoretically possible to fill the pre-write queue 22. If the pre-write queue 22 fills, it blocks any additional requests from being processed from the read queue 14. Again, this has a negligible performance impact.

While preferred embodiments have been shown and described, various modifications and substitutions may be made thereto without departing from the spirit
10 and scope of the invention. Accordingly, it is to be understood that the present invention has been described by way of illustration only, and such illustrations and embodiments as have been disclosed herein are not to be construed as limiting to the claims.

What is claimed is:

- 1 1. A method for holding up R-unit operands for a minimum number of cycles
2 until all prior updates have completed by comparing addresses in at least one queue and
3 interlocking valid R-unit register address matches, the method comprising:
4 receiving a plurality of R-unit register addresses;
5 storing said R-unit register addresses in a plurality of queues;
6 accessing said queues;
7 comparing said R-unit register addresses;
8 determining matches between R-unit register addresses; and
9 implementing one or more interlocks after said determining a valid match.
- 1 2. The method of claim 1 wherein said interlock causes a read instruction
2 not to execute.
- 1 3. The method of claim 1 wherein said plurality of queues includes a
2 write queue, pre-write queue and a read queue.
- 1 4. The method of claim 3 wherein a bypass sends an R-unit register addresses
2 when said read queue is empty.
- 1 5. The method of claim 3 wherein said comparing includes comparing
2 said R-unit register addresses sent to said read queue against said R-unit register
3 addresses sent to said write queue.

POU920000162US1

1 6. The method in claim 3 wherein said determining includes matching a
2 valid R-unit register addresses of said write queue and said read queue.

1 7. The method in claim 3 wherein said determining includes matching
2 said valid R-unit register addresses of said pre-write queue and said read queue.

1 8. The method in claim 1 wherein said interlocks are implemented after
2 said valid R-unit register address match is determined.

1 9. The method in claim 1 wherein said interlocks prevent read
2 instructions from being processed.

1 10. The method in claim 1 wherein said write queue accumulates R-unit
2 register addresses.

1 11. The method in claim 1 wherein said updating occurs when an SRAM
2 receives the accumulated results from said write queue.

1 12. A system for holding up R-unit operands for a minimum number of cycles
2 until all prior updates have completed by comparing R-unit register addresses in at least
3 one queue and interlocking valid R-unit register address matches, the system comprising:
4 a plurality of queues for storing R-unit register addresses;
5 a comparator for comparing said R-unit register addresses in said plurality
6 of queues and determining matches between R-unit register addresses; and
7 a plurality of interlocks that are implemented after determining valid
8 matches of said R-unit register addresses.

1 13. The system of claim 12 wherein one of said interlocks causes a read
2 instruction not to execute.

1 14. The system of claim 12 wherein said plurality of queues includes a
2 write queue, a pre-write queue, and a read queue.

1 15. The system of claim 14 further comprising a bypass that sends an R-unit
2 register addresses when said read queue is empty.

1 16. The system of claim 14 wherein said comparator compares
2 said R-unit register addresses sent to said read queue against said R-unit register
3 addresses sent to said write queue.

1 17. The system in claim 15 wherein said comparator determines said valid
2 R-unit register address matches between said write queue and said read queue.

1 18. The system in claim 15 wherein said comparator determines said valid
2 R-unit register address matches between said pre-write queue and said read queue.

1 19. The system in claim 13 wherein said interlocks are implemented after said
2 valid R-unit register address match is determined.

1 20. The system in claim 13 wherein said interlocks prevent read instructions
2 from being processed.

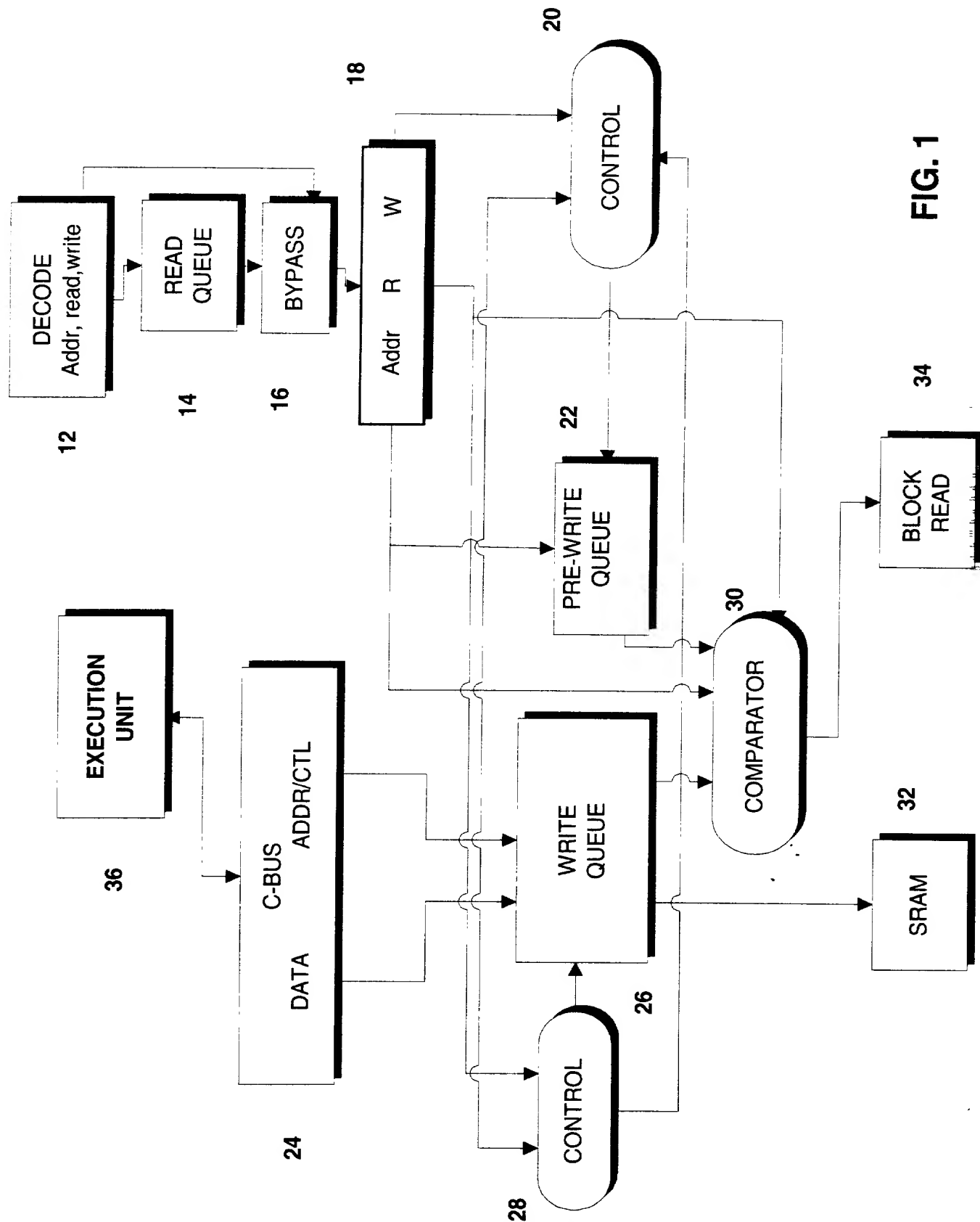
1 21. The system in claim 14 wherein said updating allows R-unit register
2 addresses to accumulate in said write queue.

1 22. The system in claim 14 wherein said R-unit is updated when an SRAM
2 receives the accumulated results from said write queue.

WRITE-BEFORE-READ INTERLOCK FOR R-UNIT OPERANDS

ABSTRACT OF THE DISCLOSURE

An exemplary embodiment of the invention is a method for holding up R-unit operands for a minimum number of cycles until all prior updates have completed by
5 comparing addresses in at least one queue and interlocking valid R-unit register address matches. The method includes receiving a plurality of R-unit register addresses and storing these R-unit register addresses in at least one queue. This method includes a write queue, a read queue, and a pre-write queue. Further, this method requires accessing these queues and comparing the R-unit register addresses therein. After the addresses are
10 compared the method determines whether there is a valid match between the R-unit register addresses and if so, implementing one or more interlocks.



Docket No.
POU920000162US1

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

WRITE-BEFORE-READ INTERLOCK FOR R-UNIT OPERANDS

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____ and was amended on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

(Number)

(Country)

(Day/Month/Year Filed)

☐

(Number)

(Country)

(Day/Month/Year Filed)

☐

(Number)

(Country)

(Day/Month/Year Filed)

☐

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

William B. Porter, Reg. No. 33,135

Philmore H. Colburn II, Reg. No. 35,101

Floyd A. Gonzalez, Reg. No. 26,732

David A. Fox, Reg. No. 38,807

Lynn L. Augspurger, Reg. No. 24,227

William A. Kinnaman, Reg. No. 27,650

Lily Neff, Reg. No. 38,254

Marc A. Ehrlich, Reg. No. 39,966

Lawrence D. Cutter, Reg. No. 28,501

Christopher A. Hughes, Reg. No. 26,914

Edward A. Pennington, Reg. No. 32,588

John E. Hoel, Reg. No. 26,279

Joseph C. Redmond, Jr., Reg. No. 18,753

Send Correspondence to: **Philmore H. Colburn II**
CANTOR COLBURN LLP
55 Griffin Road South
Bloomfield, CT 06002

Direct Telephone Calls to: *(name and telephone number)*
Philmore H. Colburn II, (860) 286-2929

Full name of sole or first inventor Scott B. Swaney	
Sole or first inventor's signature <i>Scott B. Swaney</i>	Date <i>9-29-00</i>
Residence 108 Allen Street, Catskill, NY 12414	
Citizenship United States of America	
Post Office Address Same As Residence	

Full name of second inventor, if any Mark S. Farrell	
Second inventor's signature <i>Mark S. Farrell</i>	Date <i>10/2/2000</i>
Residence 38 Stirrup Lane, Pleasant Valley, NY 12569	
Citizenship United States of America	
Post Office Address Same As Residence	

Full name of third inventor, if any Robert F. Hatch	
Third inventor's signature <i>Robert F. Hatch</i>	Date 9/28/00
Residence 241 Hardenburg Road, Ulster Park, NY 12487	
Citizenship United States of America	
Post Office Address Same As Residence	

Full name of fourth inventor, if any David P. Hillerud	
Fourth inventor's signature <i>David P. Hillerud</i>	Date 10/2/00
Residence 80 Overocker Road, Poughkeepsie, NY 12603	
Citizenship United States of America	
Post Office Address Same As Residence	

Full name of fifth inventor, if any Charles F. Webb	
Fifth inventor's signature <i>Charles F. Webb</i>	Date 9/28/2000
Residence 6 Mainetti Drive, Poughkeepsie, NY 12603	
Citizenship United States of America	
Post Office Address Same As Residence	

Full name of sixth inventor, if any	
Sixth inventor's signature	Date
Residence	
Citizenship	
Post Office Address	